Waterpixels

Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, Etienne Decencière

Waterpixels

Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, and Etienne Decencière

Abstract—**Many approaches for image segmentation rely on a first low-level segmentation step, where an image is partitioned into homogeneous regions with enforced regularity and adherence to object boundaries. Methods to generate these superpixels have gained substantial interest in the last few years, but only a few have made it into applications in practice, in particular because the requirements on the processing time are essential but are not met by most of them. Here, we propose waterpixels as a general strategy for generating superpixels which relies on the marker controlled watershed transformation. We introduce a spatially regularized gradient to achieve a tunable tradeoff between the superpixel regularity and the adherence to object boundaries. The complexity of the resulting methods is linear with respect to the number of image pixels. We quantitatively evaluate our approach on the Berkeley segmentation database and compare it against the state-of-the-art.**

Index Terms—**Superpixels, watershed, segmentation.**

I. Introduction

SUPERPIXELS (SP) are regions resulting from a low-level segmentation of an image and are typically used as primitives for further analysis such as detection, segmentation, and classification of objects (see Figure 1 for an illustration). The underlying idea is that this first low-level partition alleviates the computational complexity of the following processing steps and improves their robustness, as not single pixel values but pixel set features can be used.

Superpixels should have the following properties:

1) **homogeneity:** pixels of a given SP should present similar colors or gray levels;

Manuscript received December 18, 2014; revised April 17, 2015; accepted June 15, 2015. The work of T. Walter was supported by the European Community through the Seventh Framework Programme (FP7/2007-2013) within the Systems Microscopy under Grant 258068. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yonggang Shi.

AQ:1

AQ:2 AQ:3 V. Machairas, M. Faessel, T. Chabardes, and E. Decencière are with MINES ParisTech, Paris 75006, France, also with PSL Research University, Paris 75005, France, and also with the Center for Mathematical Morphology, Fontainebleau 77305, France (e-mail: vaia.machairas@mines-paristech.fr; matthieu.faessel@mines-paristech.fr; theodore.chabardes@mines-paristech.fr; etienne.decenciere@mines-paristech.fr).

D. Cárdenas-Peña is with the Signal Processing and Recognition Group, Universidad Nacional de Colombia, Manizales 170001-17, Colombia (e-mail: dcardenasp@unal.edu.co).

T. Walter is with MINES ParisTech, Paris 75006, France, also with PSL Research University, Paris 75005, France, also with the Centre for Computational Biology, Fontainebleau 77305, France, also with the Institut Curie, Paris 75248, France, and also with Inserm, Paris 75248, France (e-mail: thomas.walter@mines-paristech.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2451011

(a) (b)

Fig. 1. Superpixels illustration. The original image comes from the Berkeley segmentation database. (a) Original image. (b) Waterpixels.

2) **connected partition:** each SP is made of a single connected component and the SPs constitute a partition of the image;
3) **adherence to object boundaries:** object boundaries should be included in SP boundaries;
4) **regularity:** SPs should form a regular pattern on the image. This property is often desirable as it makes the SP more convenient to use for subsequent analysis steps.

The requirements on regularity and boundary adherence are to a certain extent oppositional, and a good solution typically aims at finding a compromise between these two requirements.

In addition to these requirements on superpixel quality, computational efficiency is an absolutely essential aspect, as the partition into superpixels is typically only the first step of an often complex and potentially time consuming workflow. Methods of linear complexity are consequently of particular interest.

We therefore hypothesized that the Watershed transformation [1], [2] should be an interesting candidate for superpixel generation, as it has been shown to achieve state-of-the-art performance in many segmentation problems, it is non-parametric, and there exist linear-complexity algorithms to compute it, as well as efficient implementations [3], [4]. The only often cited drawback, oversegmentation, does not seem to be problematic for superpixel generation, as long as we can control the degree of oversegmentation (number of superpixels), and the regularity of the resulting partition.

Given these considerations, we propose a strategy for applying the watershed transform to superpixel generation, where we use a spatially regularized gradient to achieve a tunable trade-off between superpixel regularity and adherence to object boundaries. We quantitatively evaluate our method on the Berkeley segmentation database and show that we outperform the best linear-time state-of-the art method: Simple Linear Iterative Clustering (SLIC) [5]. We call the resulting superpixels "waterpixels."

AQ:4

TABLE I

RECAP CHART OF EXISTING METHODS TO COMPUTE REGULAR SUPERPIXELS (n IS THE NUMBER OF PIXELS IN THE IMAGE; i IS THE NUMBER OF ITERATIONS REQUIRED; N THE NUMBER OF SUPERPIXELS). "WP" CORRESPONDS TO OUR METHOD, CALLED "WATERPIXELS"

Method	[11]	[12]	[13]	[5]	WP
Generation type (see section II-B)	1	2	1 (iterated)	2	1
Seed type (see section II-A)	A	C	C	C	B
Control on number of SPs	yes	yes	yes	no	yes
Control on regularity	no	yes	no	yes	yes
Post-processing free	no	no	no	no	yes
Complexity	$O(n)$	$O(in\sqrt{N})$	$O(in)$	$O(n)$	$O(n)$

This paper is an extended version of [6]. It proposes a more general approach (elaborating a whole family of waterpixels generation methods), with a more thorough validation and improved results with regard to the trade-off between boundary adherence and regularity, as well as computation time. Moreover, we have developed and made available a fast implementation of waterpixels.

II. RELATED WORK

Low-level segmentations have been used for a long time as first step towards segmentation [7], [8]. The term superpixel was coined much later [9], albeit in a more constrained framework. This approach has raised increasing interest since then. Various methods exist to compute SPs, most of them based on graphs [10], geometrical flows [11] or k-means [5]. We will focus on linear complexity methods generating regular SPs.

Methods for SP generation are all based on two steps: an initialization step where either seeds or a starting partition are defined and a (potentially iterative) assignment step, where each pixel is assigned to one superpixel, starting from the initialization. In the next section, we are going to review previously published approaches for SP generation with respect to these aspects and compare them regarding various performance criteria. We limit the presentation of existing methods to those with linear complexity.

A. Choosing the Seeds

In the first step, a set of seeds is chosen, which are typically spaced regularly over the image plane and which can be either regions or single pixels:

- Type A seeds are independent of the image content. These are typically the cells or the centers of a regular grid.
- Type B seeds depend on the content of the image (compromise between a regular cover of the image plane and an adaption to the contour).
- Type C seeds are initially image independent, then they are iteratively refined to take into account the image contents.

If the seed does not depend on the image, an iterative refinement is usually preferable, and therefore more time is spent on the computation of the SP. Type B methods may spend more time on finding appropriate seeds, but can therefore afford not to iterate the SP generation.

B. Building Superpixels From Seeds

In the second step, the partition into superpixels is built from the seeds. Among the methods with linear complexity, there are two main strategies for this:

Shortest Path Methods (Type 1) [11], [13]: these methods are based on region growing: they start from a set of seeds (points or regions) and successively extend them by incorporating pixels in their neighborhood according to a usually image dependent cost function until every pixel of the image plane has been assigned to exactly one superpixel. This process may or may not be iterated.

Shortest Distance Methods (Type 2) [5], [12]: these are iterative procedures inspired by the field of unsupervised learning, where at each iteration step, seeds (such as centroids) are calculated from the previous partition and pixels are then re-assigned to the closest seed (like for example the k-means approach).

Even though methods inspired by general clustering methods (type 2) seem appealing at first sight, in particular when they globally optimize a cost function, this class of methods does not guarantee connectivity of the superpixels for arbitrary choices of the pixel-seed distance (see [5], [12]). For instance, the distance metric proposed in [5] (a combination of Euclidean and grey level distance), leads to non-connected superpixels, which is undesirable. To solve this issue, a post-processing step is necessary, consisting either in relabeling the image so that every connected component has its own label (see [12]), leading to a more irregular distribution of SP sizes and shapes, or in reassigning isolated regions to the closest and large enough Superpixel, as in [5], leading to non-optimality of the solution and an unpredictable number of superpixels. In addition, such postprocessing increases the computational cost and can turn out to be the most time-consuming step when the image contains numerous small objects/details compared to the size of the Superpixel.

On the contrary, methods based on region growing (type 1) inherently implement a "path-type" distance, where the distance between two pixels does not only depend on value and position of the pixels themselves, but on values and positions along the path connecting them. Type 1 methods imply connected superpixel regions, for which the number of superpixels is exactly the number of seeds.

C. Other Properties

It is generally accepted that a good superpixel-generation method should provide to the user total control over the number of resulting Superpixels. While this property is achieved by [11]–[14], some only reach approximatively this number because of post-processing (either by splitting too big superpixels, or removing small isolated superpixels as in [5]). Another parameter is the control on superpixels regularity in the trade-off between regularity and adherence to contours. Only [5] and [12] enable the user to weight the importance of regularity compared to boundary adherence, so it can be adapted to the application.

As far as performance is concerned, one of the main criteria is undoubtedly the complexity that the method requires. Indeed, for Superpixels to be used as primitives for further analysis such as classification, their computation should neither take too long nor too much memory. This is the reason why we focus on linear complexity methods. Among them, SLIC appears to offer the best performance with regards to the trade-off between adherence to boundaries and regularity [5]. Moreover, since its recent inception, this method has become very popular in the computer vision community. We will therefore use it as reference for the quantitative evaluation of our method.

D. Superpixels and Watershed

In principle, the watershed transformation (see [15] for a review) is well suited for SP generation:

1) It gives a good adherence to object boundaries when computed on the image gradient.
2) It allows to control the number and spatial arrangement of the resulting regions through the choice of markers.
3) The connectivity of resulting regions is guaranteed and no postprocessing is required.
4) It offers linear complexity with the number of pixels in the image.

Indeed, it has been used to produce low-level segmentations in several applications, including computation intensive 3D applications [16], [17], in particular when shape regularity of the elementary regions was not required.

Previous publications claimed that the watershed transformation does not allow for the generation of spatially regular SP [5], [11]. Recently, we and others [6], [18] have shown that in principle the watershed transformation can be applied to SP generation.

Here, we introduce waterpixels, a family of methods based on the watershed transformation to compute superpixels.

III. Waterpixels

As most watershed-based segmentation methods, waterpixels are based on two steps: the definition of markers, from which the flooding starts, and the definition of a gradient (the image to be flooded). We propose to design these steps in such a way that regularity is encouraged.

A waterpixel-generation method is characterized by the following steps:

1) Computation of the gradient of the image;
2) Definition of regular cells on the image, centered on the vertices of a regular grid;
3) Selection of one marker per cell;
4) Spatial regularization of the gradient with the help of a distance function;
5) Application of the watershed transformation on the regularized gradient defined in step 4 from the markers defined in step 2.

These steps are illustrated in figure 2 and developed in the next paragraphs.

A. Gradient and Cells Definition

Let $f : D \rightarrow V$ be an image, where D is a rectangular subset of Z^2, and V a set of values, typically $\{0, \ldots, 255\}$ when f is a grey level image, or $\{0, \ldots, 255\}^3$ for color images.

The first step consists in computing the gradient image g of the image f. The choice of the gradient operator depends on the image type, *e.g.* for grey level images we might choose a morphological gradient. This gradient will be used to choose the seeds (section III-B) and to build the regularised gradient (III-C).

For the definition of cells, we first choose a set of N points $\{o_i\}_{1 \leq i \leq N}$ in D, called *cell centers*, so that they are placed on the vertices of a regular grid (a square or hexagonal one for example). Given a distance d on D, we denote by σ the grid step, *i.e.* the distance between closest grid points.

A Voronoi tesselation allows to associate to each o_i a Voronoi cell . For each such cell, a homothety centered on o_i with factor ρ $(0 < \rho \leq 1)$ leads to the computation of the final cell C_i. This last step allows for the creation of a margin between neighbouring cells, in order to avoid the selection of markers too close from each other.

B. Selection of the Markers

As each cell is meant to correspond to the generation of a unique waterpixel, our method, through the choice of one marker per cell, offers total control over the number of SP, with a strong impact on their size and shape if desired.

First, we compute the minima of the gradient g. Each minimum is a connected component, composed of one or more pixels. These minima are truncated along the grid, i.e. pixels which fall on the margins between cells are removed.

Second, every cell of the grid serves to define a region of interest in the gradient image. The content of g in this very region is then analyzed to select a unique marker, as explained in the next paragraph.

For each cell, the corresponding marker is chosen among the minima of g which are present in this very cell.

If several minima are present, then the one with the highest surface extinction value [19] is used. We have found surface extinction values to give the best performances compared with volume and dynamic extinction values (data not shown).

It may happen that there is no minimum in a cell. This is an uncommon situation in natural images. In such cases, we must add a marker for the cell which is not a minimum of g, in order to keep regularity. One solution could be to simply choose the center of the cell; however, if this point falls on a local maximum of the gradient g, the resulting SP may coincide with the maximum region and therefore be small in size (leading to a larger variability in size of the SP). We propose instead to take, as marker, the flat zone with minimum value of the gradient inside this very cell.

In both cases (i.e. either there exists at least one minimum in the cell or there is not), the selected marker has to be composed of a unique connected component to ensure regularity and connectivity of the resulting superpixel. However, it might not be the case, respectively if more than one minimum

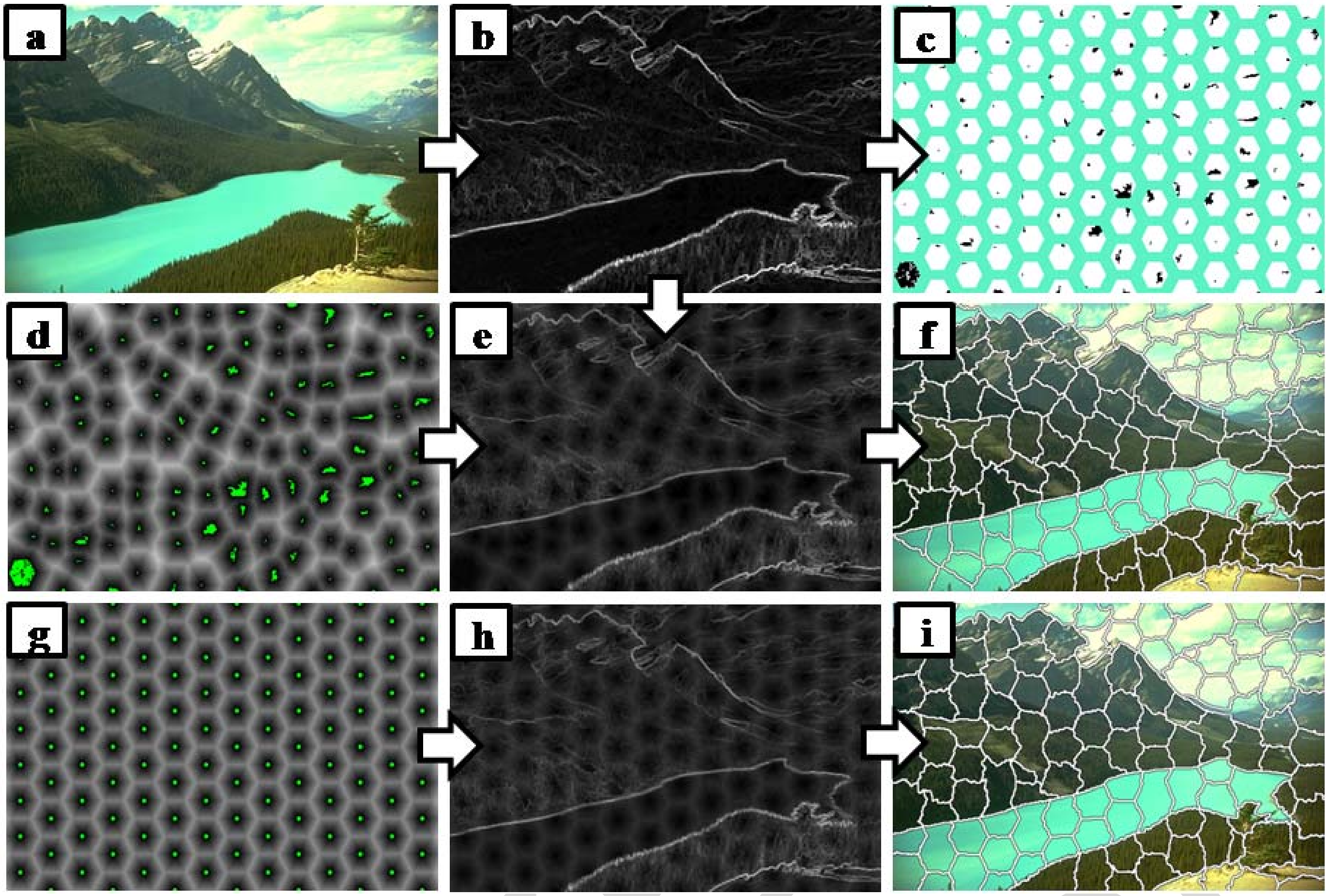


Fig. 2. Illustration of waterpixels generation: (a): original image; (b) corresponding Lab gradient; (c): selected markers within the regular grid of hexagonal cells (step $\sigma = 40$ pixels); (d): distance function to markers; (g): distance function to cell centers; (e) and (h): spatially regularized gradient respectively with distance functions to selected markers (d) and to cell centers (g); (f) and (i): Resulting waterpixels obtained by respectively applying the watershed transformation to (e) and (h), with markers (c).

have the same highest extinction value, or if more than one flat zone present the same lowest gradient value in the cell. Therefore, an additional step enables to keep only one of the connected components if there is more than one potential "best" candidate.

The set of resulting markers is denoted $\{M_i\}_{1 \geq i \geq N}$, $M_i \subset D$. The result of the marker selection procedure is illustrated in Figure 2.c.

C. Spatial Regularization of the Gradient and Watershed

The selection of markers has enforced the pertinence of future superpixel-boundaries but also the regularity of their pattern (by imposing only one marker per cell). In this paragraph, we design a spatially regularized gradient in order to further compromise between boundary adherence and regularity.

Let $Q = \{q_i\}_{1 \leq i \leq N}$ be a set of N connected components of the image f. For all $p \in D$, we can define a distance function d_Q with respect to Q as follows:

$$\forall p \in D,\ d_Q(p) = \frac{2}{\sigma} \min_{i \in [1,N]} d(p, q_i) \tag{1}$$

where σ is the grid step defined in the previous section. The normalization by σ is introduced to make the regularization independent from the chosen SP size.

We have studied two possible choices of the q_i. The first one is to choose them equal to the markers: $q_i = M_i$. Resulting waterpixels are called m-waterpixels. The second one consists in setting them at the cell centers: $q_i = o_i$, which leads to c-waterpixels. We have found that the first gives the best adherence to object boundaries, while the second produces more regular superpixels.

The spatially regularized gradient g_{reg} is defined as follows:

$$g_{reg} = g + k d_Q \tag{2}$$

where g is the gradient of the image f, d_Q is the distance function defined above and k is the spatial regularization parameter, which takes its values within $\Re^+$. The choice of k is application dependent: when k equals zero, no regularization of the gradient is applied; when $k \to \infty$, we approach the Voronoi tessellation of the set $\{q_i\}_{1 \geq i \geq N}$ in the spatial domain.

In the final step, we apply the watershed transformation on the spatially regularized gradient g_{reg}, starting the flooding from the markers $\{M_i\}_{1 \leq i \leq N}$, so that an image partition $\{s_i\}_{1 \leq i \leq N}$ is obtained. The s_i are the resulting waterpixels.

IV. Experiments

In order to evaluate waterpixels, the proposed method has been applied on the Berkeley segmentation database [20] and benchmarked against the state-of-the-art. This database is divided into three subsets, “train”, “test” and “val”, containing respectively 200, 200 and 100 images of sizes 321×481 or 481×321 pixels. Approximately 6 human-annotated ground-truth segmentations are given for each image. These ground-truth images correspond to manually drawn contours.

A. Implementation

We have found that it is beneficial to pre-process the images from the database using an area opening followed by an area closing, both of size $\sigma^2/16$ (where σ is the chosen step size of the regular grid). This operation efficiently removes details which are clearly smaller than the expected waterpixel area and which should therefore not give rise to a superpixel contour.

The Lab-gradient is adopted here in order to best reflect our visual perception of color differences and hence the pertinence of detected objects. The margin parameter ρ, described in III-A, is set to $\frac{2}{3}$.

The cell centers correspond to the vertices of a square or an hexagonal grid of step σ. The grid is computed in one pass over the image, by first calculating analytically the coordinates of the set of pixels belonging to each cell and then assigning to them the label of their corresponding cell. We will display the results for the hexagonal grid, as hexagons are more isotropic than squares. Interestingly, they also lead to a better quantitative performance, which was intuitively expected.

The implementation of the waterpixels was done using the Simple Morphological Image Library (SMIL) [21]. SMIL is a Mathematical Morphology library that aims to be fast, light-weight and portable. It brings most classical morphological operators re-designed in order to take advantage of recent computer features (SIMD, parallel processing, …) to allow handling of very large images and real time processing.

B. Qualitative Analysis

Figure 3 shows various images from the Berkeley segmentation database and their corresponding waterpixels (m-waterpixels and c-waterpixels, hexagonal and square grids, different steps). Figures 3.b and 3.c (zooms of original image presented in 3.a for m-waterpixels and c-waterpixels respectively) show the influence of the regularization parameter k (0, 4, 8, 16) for an homogeneous (blue sky) and a textured (orange rock) regions. As expected, when $k \to \infty$, m-waterpixels tend towards the Voronoi tessellation of the markers, while c-waterpixels approach the regular grid of hexagonal cells. Both show good adherence to object boundaries, as shown in Figures 3.d, 3.e, 3.f. Of course, enforcing regularity decreases the adherence to object boundaries (see the zoom in Figure 3.f for $k = 16$). One advantage of waterpixels is that the user can choose the shape (and size) of resulting superpixels depending on the application requisites. Figure 3.d, for example, presents waterpixels for hexagonal (second and third columns) and square (fourth column) grids.

As a gradient-based approach, the quality of the watershed is dependant on the borders contrast. If we look at the contours of objects missed by waterpixels, we see that it is due to the weakness of the gradient, as illustrated in Figure 4.

C. Evaluation Criteria

SP methods produce an image partition $\{s_i\}_{1 \le i \le N}$. In order to compute the SP borders, we use a morphological gradient with a 4 neighborhood. Note that the resulting contours are two pixels wide. To this set S_c, we add the one pixel wide image borders S_b. The final set is denoted C. The ground truth image corresponding to the contours of the objects to be segmented, provided in the Berkeley segmentation database, is called GT.

In superpixel generation, we look for an image decomposition into regular regions that adhere well to object boundaries. We propose to use three measures to evaluate this trade-off, namely boundary-recall, contour density and average mismatch factor, as well as computation time.

There are two levels of regularity: (1) the number of pixels required to describe the SP contours, which can be seen as a measure of complexity of individual SP, and (2) the similarity in size and shape between SP.

The first property is evaluated by the Contour Density, which is defined as the number of SP contour pixels divided by the total number of pixels in the image:

$$CD = \frac{\frac{1}{2}|S_c| + |S_b|}{|D|} \tag{3}$$

Note that $|S_c|$ is divided by 2 since contours are two-pixel-wide.

The second property, *i.e.* similarity in size and shape, is evaluated by an adapted version of the mismatch factor [22]. The mismatch factor measures the shape and size dissimilarity between two regions. Given two sets, A and B, the mismatch factor mf between them is defined as:

$$\begin{aligned} mf(A, B) :&= \frac{|A \cup B \setminus A \cap B|}{|A \cup B|} \\ &= 1 - \frac{|A \cap B|}{|A \cup B|} \end{aligned} \tag{4}$$

The mismatch factor and the Jaccard index thus sum to one. Aiming to measure the superpixel regularity, we adapted the mismatch factor to estimate the spread of size and shape distribution. Hence, the average mismatch factor MF is proposed as:

$$MF = \frac{1}{N} \sum_{i=1}^{N} mf(s_i^*, \widehat{s}^*) \tag{5}$$

where s_i^* is the centered version of superpixel s_i, and $\widehat{s}^*$ is the average centered shape of all superpixels. The complete definition of the average mismatch factor is given in Appendix.

Note that although compactness is sometimes used in superpixels evaluation (see [23]), it is a poor measurement for region regularity. For example, perfectly-rectangular regions are regular but not compact (because they are different from discs). Waterpixels can in principle tend towards differently shaped superpixels (rectangles, hexagons or other), depending on the grid and the regularization function used. Since the average mismatch factor compares each superpixel against an

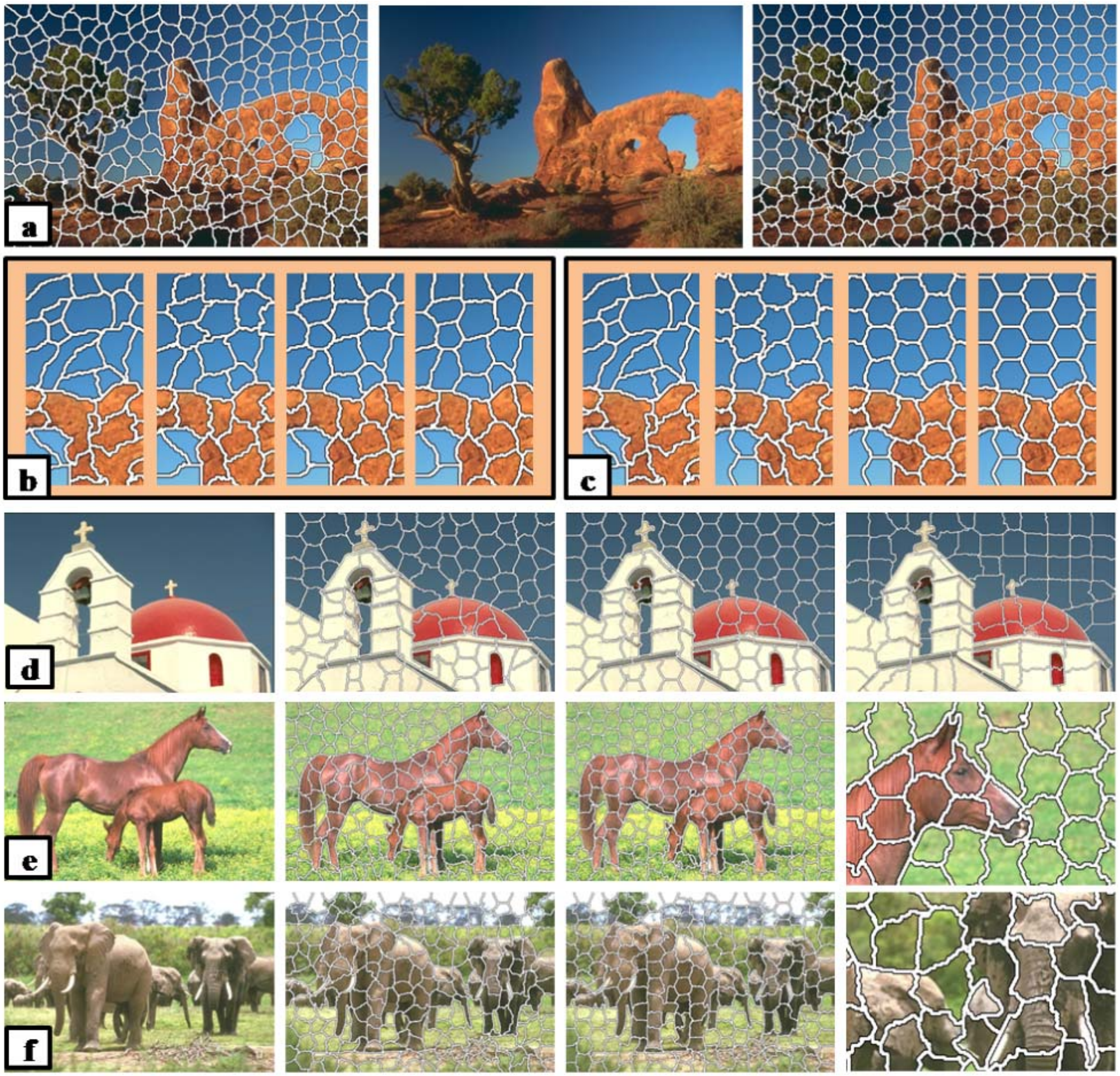


Fig. 3. Illustrations of waterpixels on the Berkeley segmentation database: All waterpixels images are computed with an hexagonal grid with step $\sigma = 30$ pixels and a regularization parameter $k = 8$, unless otherwise specified. (a): original image (middle) with corresponding *m-waterpixels* (left) and *c-waterpixels* (right). $\sigma = 25$ pixels, $k = 16$. (c): zooms of *m-waterpixels* (a) for $k = 0, 4, 8, 16$. (c): zooms of *c-waterpixels* (a) for $k = 0, 4, 8, 16$. (d): original image - *m-wat.* - *c-wat.* - *m-wat.* with square grid and $\sigma = 40$ pixels. (e): original image - *m-wat.* - *c-wat.* - zoom of *c-wat.*. (f): original image - *m-wat.* - *c-wat.* - zoom of *m-wat.* with $k = 16$.

image dependent template, this measure is more appropriate to evaluate regularity than compactness.

To quantify the adherence to object boundaries, a classical measure used in the literature is the boundary-recall (BR). Boundary-recall is defined as the percentage of ground-truth contour pixels GT which fall within strictly less than 3 pixels from superpixel boundaries C:

$$BR = \frac{|\{p \in GT, d(p, C) < 3\}|}{|GT|} \quad (6)$$

where d is the L_1 (or Manhattan) distance.

While precision cannot be directly used in the context of over-segmentations, boundary-recall has to be, in this particular case of superpixels, interpreted with caution. Indeed, as noted also by Kalinin and Sirota [24], very tortuous contours systematically lead to better performances: because of their higher number, SP contour pixels have a higher chance of matching a true contour, increasing artificially the boundary-recall. Hence, we propose to always consider the trade-off between boundary-recall and contour density to properly evaluate the adherence to object boundaries, penalizing at the same time the cost in pixels to describe SP contours.

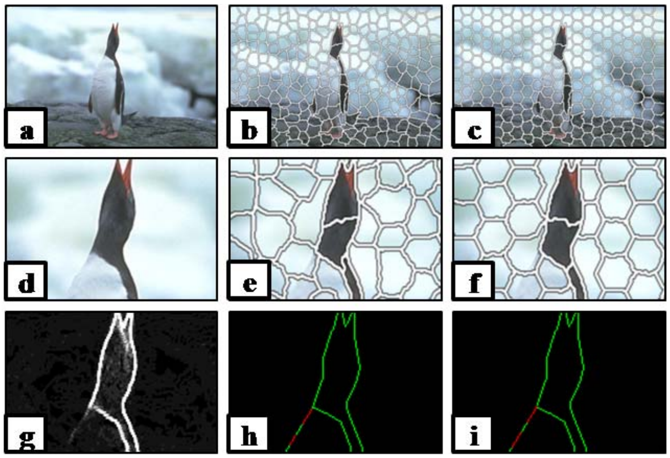


Fig. 4. Contours missed by waterpixels: (a): original image from the Berkeley segmentation database. (b): m-waterpixels with $step = 27$ and $k = 10$. (c): c-waterpixels with $step = 27$ and $k = 10$. (d), (e), (f): zoom of (a), (b), (c) respectively. (g): zoom of the non-regularized gradient image. (h) and (i): reached (green) and missed (red) contours, respectively by m-waterpixels and c-waterpixels.

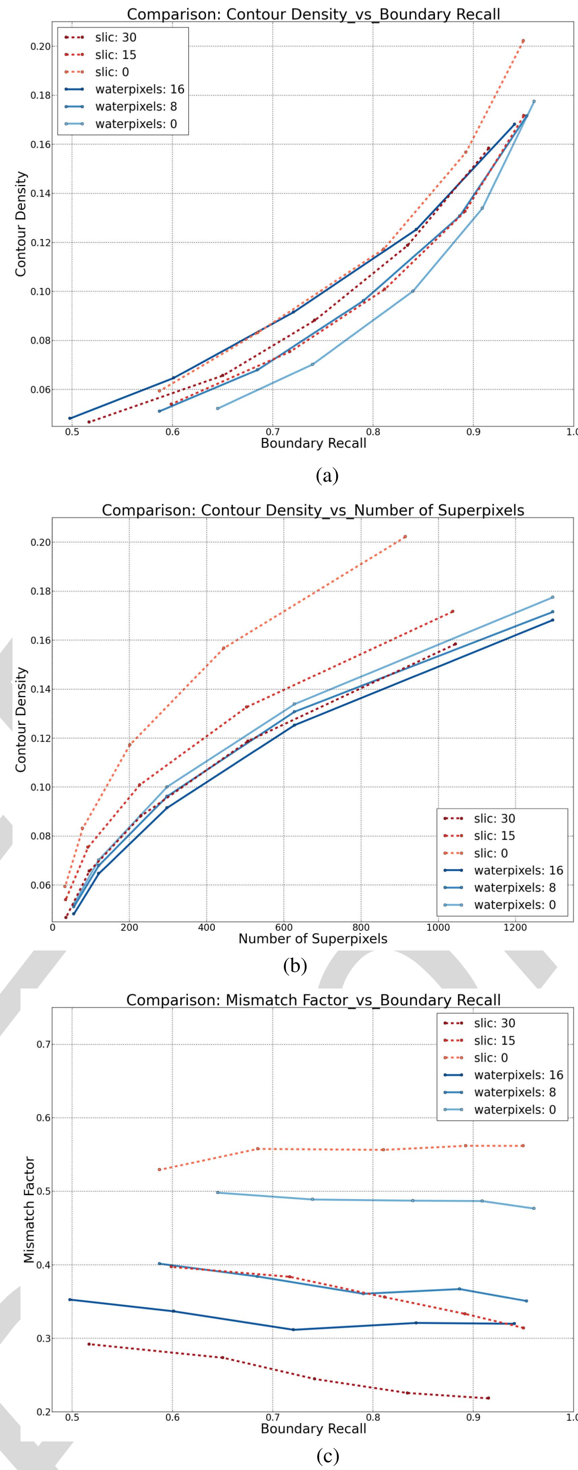


Fig. 5. Benchmark: performance comparison between waterpixels and SLIC. (a) Contour Density against Boundary-recall. (b) Contour Density against Number of Superpixels. (c) Mismatch factor against Boundary-recall.

D. Quantitative Analysis and Comparison With State-of-the-Art

In this paragraph, we will use m-waterpixels and denote them directly as "waterpixels" for the sake of simplicity.

During the design of the algorithm, we used intermediate results from the train and test subsets of the Berkeley database. Therefore, we report the results obtained for the validation subset ("val"), which contains 100 images. Results for boundary-recall, average mismatch factor and contour density are averaged for this subset and shown in Figure 5. Blue and red curves correspond to varying regularization parameters k and k' respectively for waterpixels and SLIC. The values for k and k' have been chosen such that they cover a reasonable portion of the regularization space between no regularization ($k = 0$) and a still acceptable level of regularization.

Figure 5(a) shows contour density against boundary-recall for waterpixels and SLIC. The ideal case being the lowest contour density for the highest boundary-recall, we can see that the trade-off between both properties improves for decreasing regularization, as expected. On the other hand, SLIC shows another behavior: the trade-off improves, then gets worse with regularization. At any rate, it is important to note that waterpixels achieves a better "best" trade-off than SLIC (see waterpixel $k = 0$ and SLIC $k' = 15$). Besides, this observation is valid for the whole family of waterpixel-methods as the zero-value regularization does not take into account d_Q. In order to do a fair comparison between waterpixels and SLIC over all criteria, we choose corresponding curves in the trade-off contour density/boundary-recall, *i.e.* waterpixels with $k = 8$ and SLIC with $k' = 15$, and compare this couple for the other criteria.

Figure 5(b) shows that, for a given number of superpixels, contour density of waterpixels is more stable and most of the time lower than SLIC when varying regularization. More particularly, contour density is lower for waterpixels ($k = 8$) than for SLIC ($k' = 15$). This means that for the same number of superpixels, waterpixels contours are shorter than SLIC contours, which is partly explained by less tortuous contours.

Figure 5(c) shows average mismatch factor against boundary-recall for waterpixels and SLIC. We can see that the curves for waterpixels with $k = 8$ and SLIC with $k' = 15$ are here again close to each other.

(a) (b)

Fig. 6. Comparison between Waterpixels and SLIC superpixels for $\sigma = 25$ pixels on a zoom of an image from the Berkeley segmentation database. (a) SLIC $k' = 15$. (b) Waterpixels $k = 8$.

These properties are illustrated in Figure 6, where we can see examples of reached and missed contours by both methods, as well as their different behaviours in terms of regularity (shape, size, tortuosity).

E. Computation Time

Computing time was measured on a personal computer based on Intel(R) Core(TM) i7 central processing units (4 physical cores, 4 virtual ones), operating at 2.93GHz. Both methods have linear complexity with the number of pixels in the image. For an image of size 481×321, average computing time for SLIC was 149 ms, and 132 ms for waterpixels (82 ms without pre-filtering). A more detailed comparison of computation times is presented in Figure 7 (showing average and standard deviation for different numbers of superpixels). We can see that waterpixels are generally faster to compute than SLIC superpixels. Contrary to the latter's, their computation time decreases slightly with the number of superpixels. An analysis of computation times for the different steps of waterpixels reveals that this variability is only introduced by the grid computation and the minima selection procedure. Concerning grid computation time, it rises from 2 ms for small numbers of waterpixels to 27 ms for large numbers of waterpixels. This simply means that we still have to optimize this step. Concerning the computation time of the minima selection procedure, it decreases as waterpixels become larger because of pre-filtering step. Indeed, the size of this filtering is directly proportional to the cell size. As such, resulting images contain less minima, which simplifies the selection procedure. Besides, the variance observed when we change images is explained by the fact that the difficulty of minima evaluation/computation depends on the content of each image.

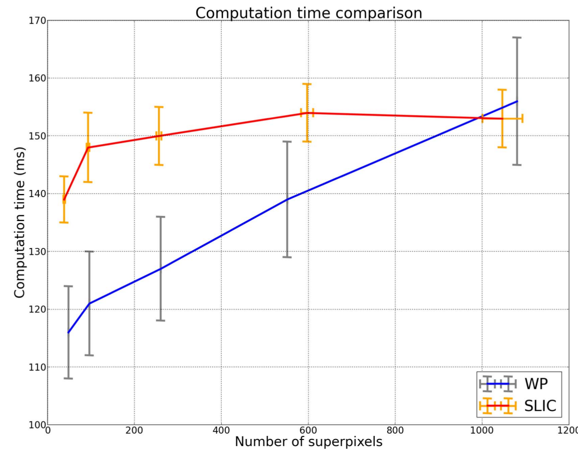


Fig. 7. Computation time comparison with images of the Berkeley database.

We are currently working on a new implementation of minima computation/evaluation which would be less dependent on the number of superpixels.

To conclude this section, waterpixels are generally faster to compute than SLIC superpixels, and they are at least as performant in the trade-off between adherence to object boundaries and regularity in shape and size, while using much less pixels to describe their contours.

V. Discussion and Perspectives

We have shown that waterpixels produce competitive results with respect to the state-of-the-art. These advantages are valuable in the classification/detection/segmentation pipeline, where superpixels play the part of primitives. Moreover, there is one major difference in the construction of the algorithm: the SLIC approach does not impose any connectivity constraint. The resulting superpixels are therefore not necessarily connected, which requires some *ad hoc* postprocessing step. In contrast, waterpixels are connected by definition, and the connectivity constraint is actually implemented in the distance used.

The proposed approach is gradient-based. Standard methods can be used to compute this gradient, or a specific gradient computation method can be designed for a given application. In any case, this offers flexibility to waterpixels. One limitation though is the quality of the signal in such a gradient image. As seen in 4, alteration by noise or insufficiently contrasted contours may lead to the prevalence of regularity over adherence to object boundaries. If filtering steps are usually enough to deal with noise and remove non pertinent small details, parameter values have to be optimized for each database. Future work will aim at overcoming this limitation by adding a learning step of optimal filtering values for specific databases.

The general design of waterpixels offers many prospects. Among them, one promising field of improvement resides in the placement of markers, as they constitute the main degree of freedom of the method. We are currently investigating the possibility to select the markers in an optimal manner, for example by formulating the marker placement as a p-dispersion problem (see [25]) in an augmented space.

The speed of waterpixels contributes to expanding their possible applications. For example, it could be interesting to compute different sets of waterpixels, by changing design options (different cells, gradients, grid steps, etc.), and then use ensemble clustering methods to obtain a final segmentation [26], [27].

Last buy not least, waterpixels lead to the efficient construction of hierarchical partitions based on superpixels. Indeed, the computation of the watershed can produce at the same time a segmentation and a hierarchy of partitions based on that segmentation, with only minor overhead computation times [28]–[30].

VI. Conclusion

This paper introduces waterpixels, a family of methods for computing regular superpixels based on the watershed transformation. Both adherence to object boundaries and regularity of resulting regions are encouraged thanks to the choice of the markers and the gradient to be flooded. Different design options, such as the distance function used to spatially regularized the gradient, lead to different trade-offs between both properties. The computational complexity of waterpixels is linear. Our current implementation makes it one of the fastest superpixel methods. Experimental results show that waterpixels are competitive with respect to the state-of-the art. They outperform SLIC superpixels, both in terms of quality and speed. The trade-off between speed and segmentation quality achieved by waterpixels, as well as their ability to generate hierarchical segmentations at negligible extra cost, offer interesting perspectives for this superpixels generation method.

An implementation of waterpixels is available from http://cmm.ensmp.fr/~machairas/waterpixels.

Appendix
Mean Mismatch Factor Definition

Let $\{s_i\}_{1 \leq i \leq N}$ be a set of superpixels. The centered version s_i^* of s_i is obtained by translating s_i so that its barycenter is the origin of the coordinates system.

The average shape $\widehat{s}^*$ of the $\{s_i\}$ is computed as follows. Let first define function S:

$$S : \left| \begin{array}{l} D \longrightarrow \mathbb{N} \\ x_p \longmapsto \sum_{i=1}^{N} 1_i(x_p) \end{array} \right. \tag{7}$$

where 1_i is the indicator function of s_i^*. Thus, image S corresponds to the summation image of all centered superpixels. Let furthermore $\mu_A = 1/n \sum_{i=1}^{N} |s_i|$ be the average area of the considered superpixels, and let S_t be the threshold of S at level t: $S_t(x) = \{x_p \in D \Big| \; \big|S(x_p)\big| \geq t\}$.

The average centered shape $\widehat{s}^*$ is then the set S_{t_0}, where t_0 is the maximal threshold value which enables $\widehat{s}^*$ to have an area greater than or equal to μ_A:

$$t_0 = \max\{t \Big| \, |S_t| \geq \mu_A\} \tag{8}$$

$$\widehat{s}^* = S_{t_0} \tag{9}$$

Finally, the mean mismatch factor of superpixels $\{s_i\}_{1 \leq i \leq N}$ is:

$$MF = \frac{1}{N} \sum_{i=1}^{N} mf(s_i^*, \widehat{s}^*). \tag{10}$$

References

[1] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection," in *Proc. Int. Workshop Image Process., Real-Time Edge Motion Detection/Estimation*, 1979. AQ:5

[2] S. Beucher and F. Meyer, "The morphological approach to segmentation: The watershed transformation," in *Mathematical Morphology in Image Processing*, E. Dougherty, Ed. 1993, pp. 433–481. AQ:6

[3] F. Meyer, "Un algorithme optimal pour la ligne de partage des eaux," *Dans 8^e Congrés Reconnaissance Formes Intell. Artif.*, vol. 2, pp. 847–857, Nov. 1991.

[4] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[5] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[6] V. Machairas, E. Decencière, and T. Walter, "Waterpixels: Superpixels based on the watershed transformation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 4343–4347.

[7] O. Monga, "An optimal region growing algorithm for image segmentation," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 1, nos. 3–4, pp. 351–375, 1987. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218001487000242

[8] B. Marcotegui and F. Meyer, "Bottom-up segmentation of image sequences for coding," *Ann. Télécommun.*, vol. 52, nos. 7–8, pp. 397–407, 1997. [Online]. Available: http://link.springer.com/article/10.1007/BF02998459

[9] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1. Oct. 2003, pp. 10–17.

[10] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[11] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[12] J. Wang and X. Wang, "VCells: Simple and efficient superpixels using edge-weighted centroidal Voronoi tessellations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1241–1247, Jun. 2012.

[13] G. Zeng, P. Wang, J. Wang, R. Gan, and H. Zha, "Structure-sensitive superpixels via geodesic distance," *Int. Conf. Comput. Vis.*, vol. 1, no. 1, pp. 1–27, 2011.

[14] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 211–224.

[15] P. Soille, *Morphological Image Analysis: Principles and Applications*. New York, NY, USA: Springer-Verlag, 2003.

[16] B. Andres, U. Köthe, M. Helmstaedter, W. Denk, and F. A. Hamprecht, "Segmentation of SBFSEM volume data of neural tissue by hierarchical classification," in *Pattern Recognition*, Berlin, Germany: Springer-Verlag, 2008, pp. 142–152.

[17] J. Stawiaski, E. Decencière, and F. Bidault, "Interactive liver tumor segmentation using graph cuts and watershed," in *Proc. MICCAI*, New York, NY, USA, 2008. AQ:7

[18] P. Neubert and P. Protzel, "Compact watershed and preemptive SLIC: On improving trade-offs of superpixel segmentation algorithms," in *Proc. IEEE 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 996–1001.

[19] C. Vachier and F. Meyer, "Extinction values: A new measurement of persistence," in *Proc. IEEE Workshop Non Linear Signal/Image Process.*, 1995, pp. 254–257.

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, vol. 2. Jul. 2001, pp. 416–423.

[21] M. Faessel and M. Bilodeau, "SMIL: Simple morphological image library," LRDE, Tech. Rep., 2013. AQ:8

[22] N. J. C. Strachan, P. Nesvadba, and A. R. Allen, "Fish species recognition by shape analysis of images," *Pattern Recognit.*, vol. 23, no. 5, pp. 539–544, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/003132039090074U

[23] A. Schick, M. Fischer, and R. Stiefelhagen, "An evaluation of the compactness of superpixels," *Pattern Recognit. Lett.*, vol. 43, pp. 71–80, Jul. 2014.

[24] P. Kalinin and A. Sirota, "A graph based approach to hierarchical image over-segmentation," *Comput. Vis. Image Understand.*, vol. 130, pp. 80–86, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314214001891

[25] E. Erkut, "The discrete *p*-dispersion problem," *Eur. J. Oper. Res.*, vol. 46, no. 1, pp. 48–60, May 1990.

[26] K. Cho and P. Meer, "Image segmentation from consensus information," *Comput. Vis. Image Understand.*, vol. 68, no. 1, pp. 72–89, Oct. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314297905464

[27] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003. [Online]. Available: http://dx.doi.org/10.1162/153244303321897735

[28] F. Meyer, "Minimum spanning forests for morphological segmentation," in *Mathematical Morphology and Its Applications to Image Processing*. Boston, MA, USA: Kluwer, Sep. 1994, pp. 77–84.

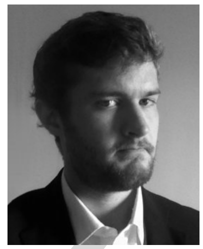
[29] S. Beucher, "Watershed, hierarchical segmentation and waterfall algorithm," in *Mathematical Morphology and Its Applications to Image Processing*, J. Serra and P. Soille, Eds. Fontainebleau, France: Kluwer, Sep. 1994, pp. 69–76.

[30] F. Meyer, "An overview of morphological segmentation," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 7, pp. 1089–1118, 2001.

Vaïa Machairas received the Engineering degree in optics from the Institut d'Optique Graduate School (Supoptique), Palaiseau, France, and the master's degree in optics, image, vision from Jean Monnet University, Saint Etienne, France, both in 2013. She is currently pursuing the Ph.D. degree with the Centre for Mathematical Morphology, MINES ParisTech. Her research interests include mathematical morphology, image segmentation, machine learning, and colorimetry.

Matthieu Faessel received the Ph.D. degree in engineer sciences from the University of Bordeaux, France, in 2003. He is currently a Research Engineer with the Centre of Mathematical Morphology, School of Mines, Paris, France. His research interests include image segmentation, computer vision, and materials.

David Cárdenas-Peña received the bachelor's degree in electronic engineering and the M.Eng. degree in industrial automation from the Universidad Nacional de Colombia, Manizales-Colombia, in 2008 and 2011, respectively. He is currently pursuing the Ph.D. degree in automatics with the Universidad Nacional de Colombia. He has been a Research Assistant with the Signal Processing and Recongnition Group since 2008. His current research interests include machine learning and signal and image processing.



Théodore Chabardes received the degree from the Engineering School, ESIEE Paris, France, in 2014, as an Engineer specialized in computer science. He is currently pursuing the Ph.D. degree with the Centre of Mathematical Morphology, School of Mines, Paris, France. His research interests include mathematical morphology, image segmentation, and software optimization.

Thomas Walter received the Diploma degree in electrical engineering from Saarland University, Germany, and the Ph.D. degree in mathematical morphology from Mines ParisTech, France. He held a post-doctoral position with the European Molecular Biology Laboratory, Heidelberg, Germany. He is currently a Team Leader in bioimage informatics with the Centre for Computational Biology, Mines ParisTech, and a member of the Bioinformatics Unit with the Curie Institute, Paris. His most visible scientific contributions have been in the field of bioimage informatics, and in particular, in high content screening. He has pioneered methods in the field of cellular phenotyping and phenotypic clustering from live cell-imaging data. He was involved in the first genome-wide screen by live cell imaging in a human cell line, and co-develops the open-source software cellcognition.

Etienne Decencière received the Engineering degree and the Ph.D. degree in mathematical morphology from MINES ParisTech, France, in 1994 and 1997, respectively, and the Habilitation à Diriger des Recherches from Jean Monnet University, in 2008. He holds a research fellow position with the Centre for Mathematical Morphology, MINES ParisTech, where he leads several academic and industrial research projects. His main research interests are in mathematical morphology, image segmentation, and biomedical applications.

AUTHOR QUERIES

AQ:1 = Please check whether the edits made in the financial section are OK.
AQ:2 = Please confirm the current affiliation of all authors.
AQ:3 = Please confirm the postal code for "MINES ParisTech, PSL Research University, Center for Mathematical Morphology, Universidad Nacional de Colombia, Centre for Computational Biology, Institut Curie, and Inserm."
AQ:4 = Table I is not cited in body text. Please indicate where it should be cited.
AQ:5 = Please provide the page range for ref. [1].
AQ:6 = Please provide the publisher name and location for ref. [2].
AQ:7 = Please provide the page range and also confirm the conference title for ref. [17].
AQ:8 = Please provide the organization, location, and report no. for ref. [21].

Waterpixels

Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, and Etienne Decencière

Abstract—**Many approaches for image segmentation rely on a first low-level segmentation step, where an image is partitioned into homogeneous regions with enforced regularity and adherence to object boundaries. Methods to generate these superpixels have gained substantial interest in the last few years, but only a few have made it into applications in practice, in particular because the requirements on the processing time are essential but are not met by most of them. Here, we propose waterpixels as a general strategy for generating superpixels which relies on the marker controlled watershed transformation. We introduce a spatially regularized gradient to achieve a tunable tradeoff between the superpixel regularity and the adherence to object boundaries. The complexity of the resulting methods is linear with respect to the number of image pixels. We quantitatively evaluate our approach on the Berkeley segmentation database and compare it against the state-of-the-art.**

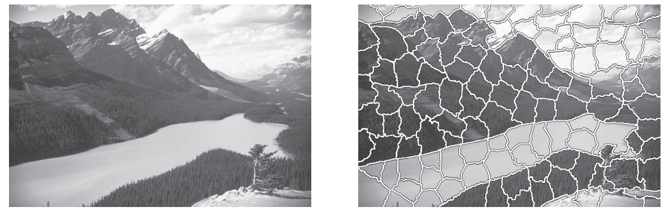
Index Terms—**Superpixels, watershed, segmentation.**

I. INTRODUCTION

SUPERPIXELS (SP) are regions resulting from a low-level segmentation of an image and are typically used as primitives for further analysis such as detection, segmentation, and classification of objects (see Figure 1 for an illustration). The underlying idea is that this first low-level partition alleviates the computational complexity of the following processing steps and improves their robustness, as not single pixel values but pixel set features can be used.

Superpixels should have the following properties:

1) **homogeneity:** pixels of a given SP should present similar colors or gray levels;



(a) (b)

Fig. 1. Superpixels illustration. The original image comes from the Berkeley segmentation database. (a) Original image. (b) Waterpixels.

2) **connected partition:** each SP is made of a single connected component and the SPs constitute a partition of the image;
3) **adherence to object boundaries:** object boundaries should be included in SP boundaries;
4) **regularity:** SPs should form a regular pattern on the image. This property is often desirable as it makes the SP more convenient to use for subsequent analysis steps.

The requirements on regularity and boundary adherence are to a certain extent oppositional, and a good solution typically aims at finding a compromise between these two requirements.

In addition to these requirements on superpixel quality, computational efficiency is an absolutely essential aspect, as the partition into superpixels is typically only the first step of an often complex and potentially time consuming workflow. Methods of linear complexity are consequently of particular interest.

We therefore hypothesized that the Watershed transformation [1], [2] should be an interesting candidate for superpixel generation, as it has been shown to achieve state-of-the-art performance in many segmentation problems, it is non-parametric, and there exist linear-complexity algorithms to compute it, as well as efficient implementations [3], [4]. The only often cited drawback, oversegmentation, does not seem to be problematic for superpixel generation, as long as we can control the degree of oversegmentation (number of superpixels), and the regularity of the resulting partition.

Given these considerations, we propose a strategy for applying the watershed transform to superpixel generation, where we use a spatially regularized gradient to achieve a tunable trade-off between superpixel regularity and adherence to object boundaries. We quantitatively evaluate our method on the Berkeley segmentation database and show that we outperform the best linear-time state-of-the art method: Simple Linear Iterative Clustering (SLIC) [5]. We call the resulting superpixels "waterpixels."

Manuscript received December 18, 2014; revised April 17, 2015; accepted
AQ:1 June 15, 2015. The work of T. Walter was supported by the European Community through the Seventh Framework Programme (FP7/2007-2013) within the Systems Microscopy under Grant 258068. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yonggang Shi.

AQ:2 AQ:3 V. Machairas, M. Faessel, T. Chabardes, and E. Decencière are with MINES ParisTech, Paris 75006, France, also with PSL Research University, Paris 75005, France, and also with the Center for Mathematical Morphology, Fontainebleau 77305, France (e-mail: vaia.machairas@mines-paristech.fr; matthieu.faessel@mines-paristech.fr; theodore.chabardes@mines-paristech.fr; etienne.decenciere@mines-paristech.fr).

D. Cárdenas-Peña is with the Signal Processing and Recognition Group, Universidad Nacional de Colombia, Manizales 170001-17, Colombia (e-mail: dcardenasp@unal.edu.co).

T. Walter is with MINES ParisTech, Paris 75006, France, also with PSL Research University, Paris 75005, France, also with the Centre for Computational Biology, Fontainebleau 77305, France, also with the Institut Curie, Paris 75248, France, and also with Inserm, Paris 75248, France (e-mail: thomas.walter@mines-paristech.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2451011

TABLE I

AQ:4

RECAP CHART OF EXISTING METHODS TO COMPUTE REGULAR SUPERPIXELS (n IS THE NUMBER OF PIXELS IN THE IMAGE; i IS THE NUMBER OF ITERATIONS REQUIRED; N THE NUMBER OF SUPERPIXELS). "WP" CORRESPONDS TO OUR METHOD, CALLED "WATERPIXELS"

Method	[11]	[12]	[13]	[5]	WP
Generation type (see section II-B)	1	2	1 (iterated)	2	1
Seed type (see section II-A)	A	C	C	C	B
Control on number of SPs	yes	yes	yes	no	yes
Control on regularity	no	yes	no	yes	yes
Post-processing free	no	no	no	no	yes
Complexity	$O(n)$	$O(in\sqrt{N})$	$O(in)$	$O(n)$	$O(n)$

This paper is an extended version of [6]. It proposes a more general approach (elaborating a whole family of waterpixels generation methods), with a more thorough validation and improved results with regard to the trade-off between boundary adherence and regularity, as well as computation time. Moreover, we have developed and made available a fast implementation of waterpixels.

II. Related Work

Low-level segmentations have been used for a long time as first step towards segmentation [7], [8]. The term superpixel was coined much later [9], albeit in a more constrained framework. This approach has raised increasing interest since then. Various methods exist to compute SPs, most of them based on graphs [10], geometrical flows [11] or k-means [5]. We will focus on linear complexity methods generating regular SPs.

Methods for SP generation are all based on two steps: an initialization step where either seeds or a starting partition are defined and a (potentially iterative) assignment step, where each pixel is assigned to one superpixel, starting from the initialization. In the next section, we are going to review previously published approaches for SP generation with respect to these aspects and compare them regarding various performance criteria. We limit the presentation of existing methods to those with linear complexity.

A. Choosing the Seeds

In the first step, a set of seeds is chosen, which are typically spaced regularly over the image plane and which can be either regions or single pixels:

- Type A seeds are independent of the image content. These are typically the cells or the centers of a regular grid.
- Type B seeds depend on the content of the image (compromise between a regular cover of the image plane and an adaption to the contour).
- Type C seeds are initially image independent, then they are iteratively refined to take into account the image contents.

If the seed does not depend on the image, an iterative refinement is usually preferable, and therefore more time is spent on the computation of the SP. Type B methods may spend more time on finding appropriate seeds, but can therefore afford not to iterate the SP generation.

B. Building Superpixels From Seeds

In the second step, the partition into superpixels is built from the seeds. Among the methods with linear complexity, there are two main strategies for this:

Shortest Path Methods (Type 1) [11], [13]: these methods are based on region growing: they start from a set of seeds (points or regions) and successively extend them by incorporating pixels in their neighborhood according to a usually image dependent cost function until every pixel of the image plane has been assigned to exactly one superpixel. This process may or may not be iterated.

Shortest Distance Methods (Type 2) [5], [12]: these are iterative procedures inspired by the field of unsupervised learning, where at each iteration step, seeds (such as centroids) are calculated from the previous partition and pixels are then re-assigned to the closest seed (like for example the k-means approach).

Even though methods inspired by general clustering methods (type 2) seem appealing at first sight, in particular when they globally optimize a cost function, this class of methods does not guarantee connectivity of the superpixels for arbitrary choices of the pixel-seed distance (see [5], [12]). For instance, the distance metric proposed in [5] (a combination of Euclidean and grey level distance), leads to non-connected superpixels, which is undesirable. To solve this issue, a post-processing step is necessary, consisting either in relabeling the image so that every connected component has its own label (see [12]), leading to a more irregular distribution of SP sizes and shapes, or in reassigning isolated regions to the closest and large enough Superpixel, as in [5], leading to non-optimality of the solution and an unpredictable number of superpixels. In addition, such postprocessing increases the computational cost and can turn out to be the most time-consuming step when the image contains numerous small objects/details compared to the size of the Superpixel.

On the contrary, methods based on region growing (type 1) inherently implement a "path-type" distance, where the distance between two pixels does not only depend on value and position of the pixels themselves, but on values and positions along the path connecting them. Type 1 methods imply connected superpixel regions, for which the number of superpixels is exactly the number of seeds.

C. Other Properties

It is generally accepted that a good superpixel-generation method should provide to the user total control over the number of resulting Superpixels. While this property is achieved by [11]–[14], some only reach approximatively this number because of post-processing (either by splitting too big superpixels, or removing small isolated superpixels as in [5]). Another parameter is the control on superpixels regularity in the trade-off between regularity and adherence to contours. Only [5] and [12] enable the user to weight the importance of regularity compared to boundary adherence, so it can be adapted to the application.

As far as performance is concerned, one of the main criteria is undoubtedly the complexity that the method requires. Indeed, for Superpixels to be used as primitives for further analysis such as classification, their computation should neither take too long nor too much memory. This is the reason why we focus on linear complexity methods. Among them, SLIC appears to offer the best performance with regards to the trade-off between adherence to boundaries and regularity [5]. Moreover, since its recent inception, this method has become very popular in the computer vision community. We will therefore use it as reference for the quantitative evaluation of our method.

D. Superpixels and Watershed

In principle, the watershed transformation (see [15] for a review) is well suited for SP generation:

1) It gives a good adherence to object boundaries when computed on the image gradient.
2) It allows to control the number and spatial arrangement of the resulting regions through the choice of markers.
3) The connectivity of resulting regions is guaranteed and no postprocessing is required.
4) It offers linear complexity with the number of pixels in the image.

Indeed, it has been used to produce low-level segmentations in several applications, including computation intensive 3D applications [16], [17], in particular when shape regularity of the elementary regions was not required.

Previous publications claimed that the watershed transformation does not allow for the generation of spatially regular SP [5], [11]. Recently, we and others [6], [18] have shown that in principle the watershed transformation can be applied to SP generation.

Here, we introduce waterpixels, a family of methods based on the watershed transformation to compute superpixels.

III. Waterpixels

As most watershed-based segmentation methods, waterpixels are based on two steps: the definition of markers, from which the flooding starts, and the definition of a gradient (the image to be flooded). We propose to design these steps in such a way that regularity is encouraged.

A waterpixel-generation method is characterized by the following steps:

1) Computation of the gradient of the image;
2) Definition of regular cells on the image, centered on the vertices of a regular grid;
3) Selection of one marker per cell;
4) Spatial regularization of the gradient with the help of a distance function;
5) Application of the watershed transformation on the regularized gradient defined in step 4 from the markers defined in step 2.

These steps are illustrated in figure 2 and developed in the next paragraphs.

A. Gradient and Cells Definition

Let $f : D \rightarrow V$ be an image, where D is a rectangular subset of Z^2, and V a set of values, typically $\{0, \ldots, 255\}$ when f is a grey level image, or $\{0, \ldots, 255\}^3$ for color images.

The first step consists in computing the gradient image g of the image f. The choice of the gradient operator depends on the image type, *e.g.* for grey level images we might choose a morphological gradient. This gradient will be used to choose the seeds (section III-B) and to build the regularised gradient (III-C).

For the definition of cells, we first choose a set of N points $\{o_i\}_{1 \leq i \leq N}$ in D, called *cell centers*, so that they are placed on the vertices of a regular grid (a square or hexagonal one for example). Given a distance d on D, we denote by σ the grid step, *i.e.* the distance between closest grid points.

A Voronoi tesselation allows to associate to each o_i a Voronoi cell . For each such cell, a homothety centered on o_i with factor ρ $(0 < \rho \leq 1)$ leads to the computation of the final cell C_i. This last step allows for the creation of a margin between neighbouring cells, in order to avoid the selection of markers too close from each other.

B. Selection of the Markers

As each cell is meant to correspond to the generation of a unique waterpixel, our method, through the choice of one marker per cell, offers total control over the number of SP, with a strong impact on their size and shape if desired.

First, we compute the minima of the gradient g. Each minimum is a connected component, composed of one or more pixels. These minima are truncated along the grid, i.e. pixels which fall on the margins between cells are removed.

Second, every cell of the grid serves to define a region of interest in the gradient image. The content of g in this very region is then analyzed to select a unique marker, as explained in the next paragraph.

For each cell, the corresponding marker is chosen among the minima of g which are present in this very cell.

If several minima are present, then the one with the highest surface extinction value [19] is used. We have found surface extinction values to give the best performances compared with volume and dynamic extinction values (data not shown).

It may happen that there is no minimum in a cell. This is an uncommon situation in natural images. In such cases, we must add a marker for the cell which is not a minimum of g, in order to keep regularity. One solution could be to simply choose the center of the cell; however, if this point falls on a local maximum of the gradient g, the resulting SP may coincide with the maximum region and therefore be small in size (leading to a larger variability in size of the SP). We propose instead to take, as marker, the flat zone with minimum value of the gradient inside this very cell.

In both cases (i.e. either there exists at least one minimum in the cell or there is not), the selected marker has to be composed of a unique connected component to ensure regularity and connectivity of the resulting superpixel. However, it might not be the case, respectively if more than one minimum

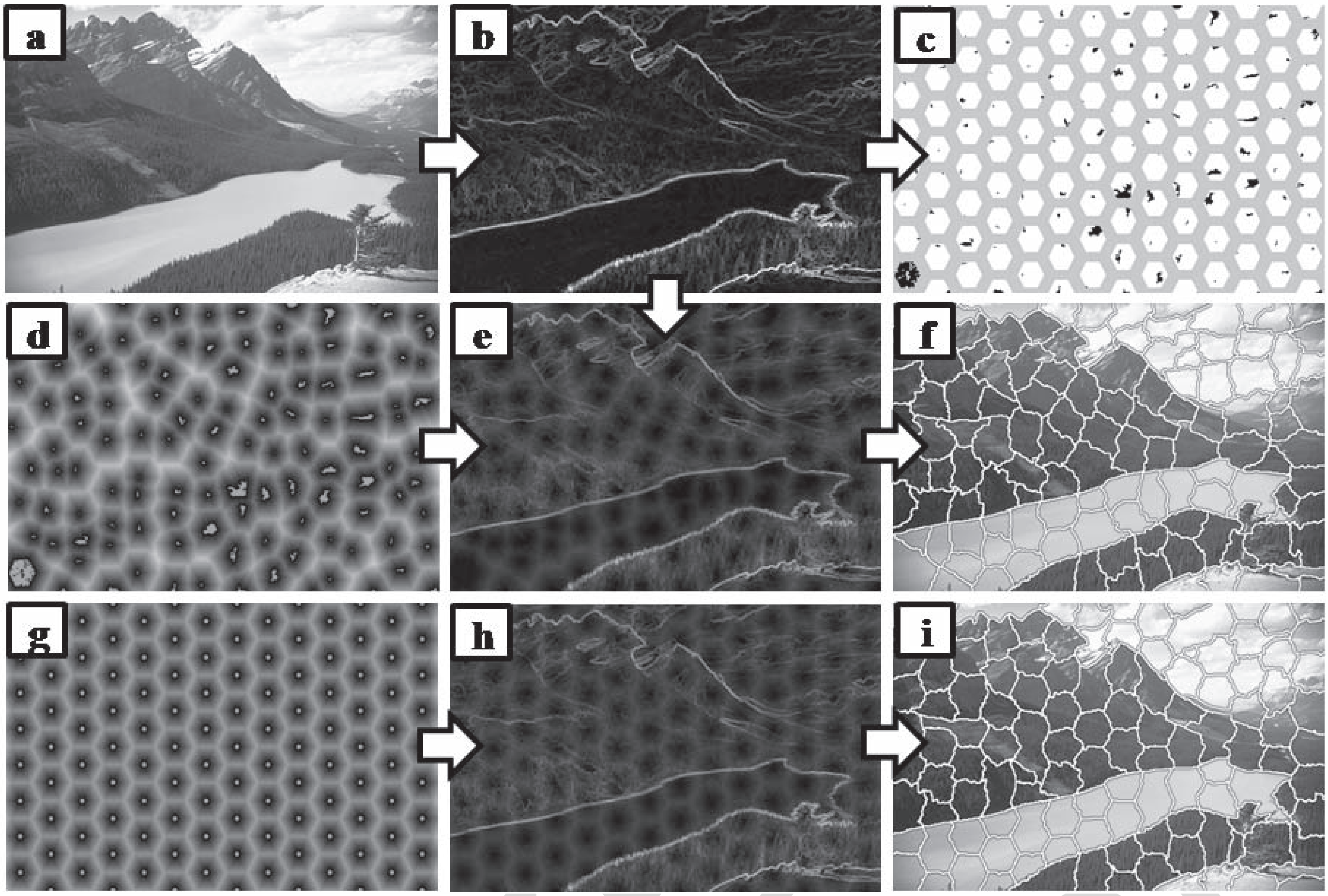


Fig. 2. Illustration of waterpixels generation: (a): original image; (b) corresponding Lab gradient; (c): selected markers within the regular grid of hexagonal cells (step $\sigma = 40$ pixels); (d): distance function to markers; (g): distance function to cell centers; (e) and (h): spatially regularized gradient respectively with distance functions to selected markers (d) and to cell centers (g); (f) and (i): Resulting waterpixels obtained by respectively applying the watershed transformation to (e) and (h), with markers (c).

have the same highest extinction value, or if more than one flat zone present the same lowest gradient value in the cell. Therefore, an additional step enables to keep only one of the connected components if there is more than one potential "best" candidate.

The set of resulting markers is denoted $\{M_i\}_{1\geq i\geq N}$, $M_i \subset D$. The result of the marker selection procedure is illustrated in Figure 2.c.

C. Spatial Regularization of the Gradient and Watershed

The selection of markers has enforced the pertinence of future superpixel-boundaries but also the regularity of their pattern (by imposing only one marker per cell). In this paragraph, we design a spatially regularized gradient in order to further compromise between boundary adherence and regularity.

Let $Q = \{q_i\}_{1\leq i\leq N}$ be a set of N connected components of the image f. For all $p \in D$, we can define a distance function d_Q with respect to Q as follows:

$$\forall p \in D,\ d_Q(p) = \frac{2}{\sigma} \min_{i\in[1,N]} d(p, q_i) \tag{1}$$

where σ is the grid step defined in the previous section. The normalization by σ is introduced to make the regularization independent from the chosen SP size.

We have studied two possible choices of the q_i. The first one is to choose them equal to the markers: $q_i = M_i$. Resulting waterpixels are called m-waterpixels. The second one consists in setting them at the cell centers: $q_i = o_i$, which leads to c-waterpixels. We have found that the first gives the best adherence to object boundaries, while the second produces more regular superpixels.

The spatially regularized gradient g_{reg} is defined as follows:

$$g_{reg} = g + k d_Q \tag{2}$$

where g is the gradient of the image f, d_Q is the distance function defined above and k is the spatial regularization parameter, which takes its values within $\Re^+$. The choice of k is application dependent: when k equals zero, no regularization of the gradient is applied; when $k \to \infty$, we approach the Voronoi tessellation of the set $\{q_i\}_{1\geq i\geq N}$ in the spatial domain.

In the final step, we apply the watershed transformation on the spatially regularized gradient g_{reg}, starting the flooding from the markers $\{M_i\}_{1\leq i\leq N}$, so that an image partition $\{s_i\}_{1\leq i\leq N}$ is obtained. The s_i are the resulting waterpixels.

IV. Experiments

In order to evaluate waterpixels, the proposed method has been applied on the Berkeley segmentation database [20] and benchmarked against the state-of-the-art. This database is divided into three subsets, "train", "test" and "val", containing respectively 200, 200 and 100 images of sizes 321×481 or 481×321 pixels. Approximately 6 human-annotated ground-truth segmentations are given for each image. These ground-truth images correspond to manually drawn contours.

A. Implementation

We have found that it is beneficial to pre-process the images from the database using an area opening followed by an area closing, both of size $\sigma^2/16$ (where σ is the chosen step size of the regular grid). This operation efficiently removes details which are clearly smaller than the expected waterpixel area and which should therefore not give rise to a superpixel contour.

The Lab-gradient is adopted here in order to best reflect our visual perception of color differences and hence the pertinence of detected objects. The margin parameter ρ, described in III-A, is set to $\frac{2}{3}$.

The cell centers correspond to the vertices of a square or an hexagonal grid of step σ. The grid is computed in one pass over the image, by first calculating analytically the coordinates of the set of pixels belonging to each cell and then assigning to them the label of their corresponding cell. We will display the results for the hexagonal grid, as hexagons are more isotropic than squares. Interestingly, they also lead to a better quantitative performance, which was intuitively expected.

The implementation of the waterpixels was done using the Simple Morphological Image Library (SMIL) [21]. SMIL is a Mathematical Morphology library that aims to be fast, lightweight and portable. It brings most classical morphological operators re-designed in order to take advantage of recent computer features (SIMD, parallel processing, …) to allow handling of very large images and real time processing.

B. Qualitative Analysis

Figure 3 shows various images from the Berkeley segmentation database and their corresponding waterpixels (m-waterpixels and c-waterpixels, hexagonal and square grids, different steps). Figures 3.b and 3.c (zooms of original image presented in 3.a for m-waterpixels and c-waterpixels respectively) show the influence of the regularization parameter k (0, 4, 8, 16) for an homogeneous (blue sky) and a textured (orange rock) regions. As expected, when $k \to \infty$, m-waterpixels tend towards the Voronoi tessellation of the markers, while c-waterpixels approach the regular grid of hexagonal cells. Both show good adherence to object boundaries, as shown in Figures 3.d, 3.e, 3.f. Of course, enforcing regularity decreases the adherence to object boundaries (see the zoom in Figure 3.f for $k = 16$). One advantage of waterpixels is that the user can choose the shape (and size) of resulting superpixels depending on the application requisites. Figure 3.d, for example, presents waterpixels for hexagonal (second and third columns) and square (fourth column) grids.

As a gradient-based approach, the quality of the watershed is dependant on the borders contrast. If we look at the contours of objects missed by waterpixels, we see that it is due to the weakness of the gradient, as illustrated in Figure 4.

C. Evaluation Criteria

SP methods produce an image partition $\{s_i\}_{1 \leq i \leq N}$. In order to compute the SP borders, we use a morphological gradient with a 4 neighborhood. Note that the resulting contours are two pixels wide. To this set S_c, we add the one pixel wide image borders S_b. The final set is denoted C. The ground truth image corresponding to the contours of the objects to be segmented, provided in the Berkeley segmentation database, is called GT.

In superpixel generation, we look for an image decomposition into regular regions that adhere well to object boundaries. We propose to use three measures to evaluate this trade-off, namely boundary-recall, contour density and average mismatch factor, as well as computation time.

There are two levels of regularity: (1) the number of pixels required to describe the SP contours, which can be seen as a measure of complexity of individual SP, and (2) the similarity in size and shape between SP.

The first property is evaluated by the Contour Density, which is defined as the number of SP contour pixels divided by the total number of pixels in the image:

$$CD = \frac{\frac{1}{2}|S_c| + |S_b|}{|D|} \tag{3}$$

Note that $|S_c|$ is divided by 2 since contours are two-pixel-wide.

The second property, *i.e.* similarity in size and shape, is evaluated by an adapted version of the mismatch factor [22]. The mismatch factor measures the shape and size dissimilarity between two regions. Given two sets, A and B, the mismatch factor mf between them is defined as:

$$\begin{aligned} mf(A, B) &:= \frac{|A \cup B \setminus A \cap B|}{|A \cup B|} \\ &= 1 - \frac{|A \cap B|}{|A \cup B|} \end{aligned} \tag{4}$$

The mismatch factor and the Jaccard index thus sum to one. Aiming to measure the superpixel regularity, we adapted the mismatch factor to estimate the spread of size and shape distribution. Hence, the average mismatch factor MF is proposed as:

$$MF = \frac{1}{N} \sum_{i=1}^{N} mf(s_i^*, \widehat{s}^*) \tag{5}$$

where s_i^* is the centered version of superpixel s_i, and $\widehat{s}^*$ is the average centered shape of all superpixels. The complete definition of the average mismatch factor is given in Appendix.

Note that although compactness is sometimes used in superpixels evaluation (see [23]), it is a poor measurement for region regularity. For example, perfectly-rectangular regions are regular but not compact (because they are different from discs). Waterpixels can in principle tend towards differently shaped superpixels (rectangles, hexagons or other), depending on the grid and the regularization function used. Since the average mismatch factor compares each superpixel against an

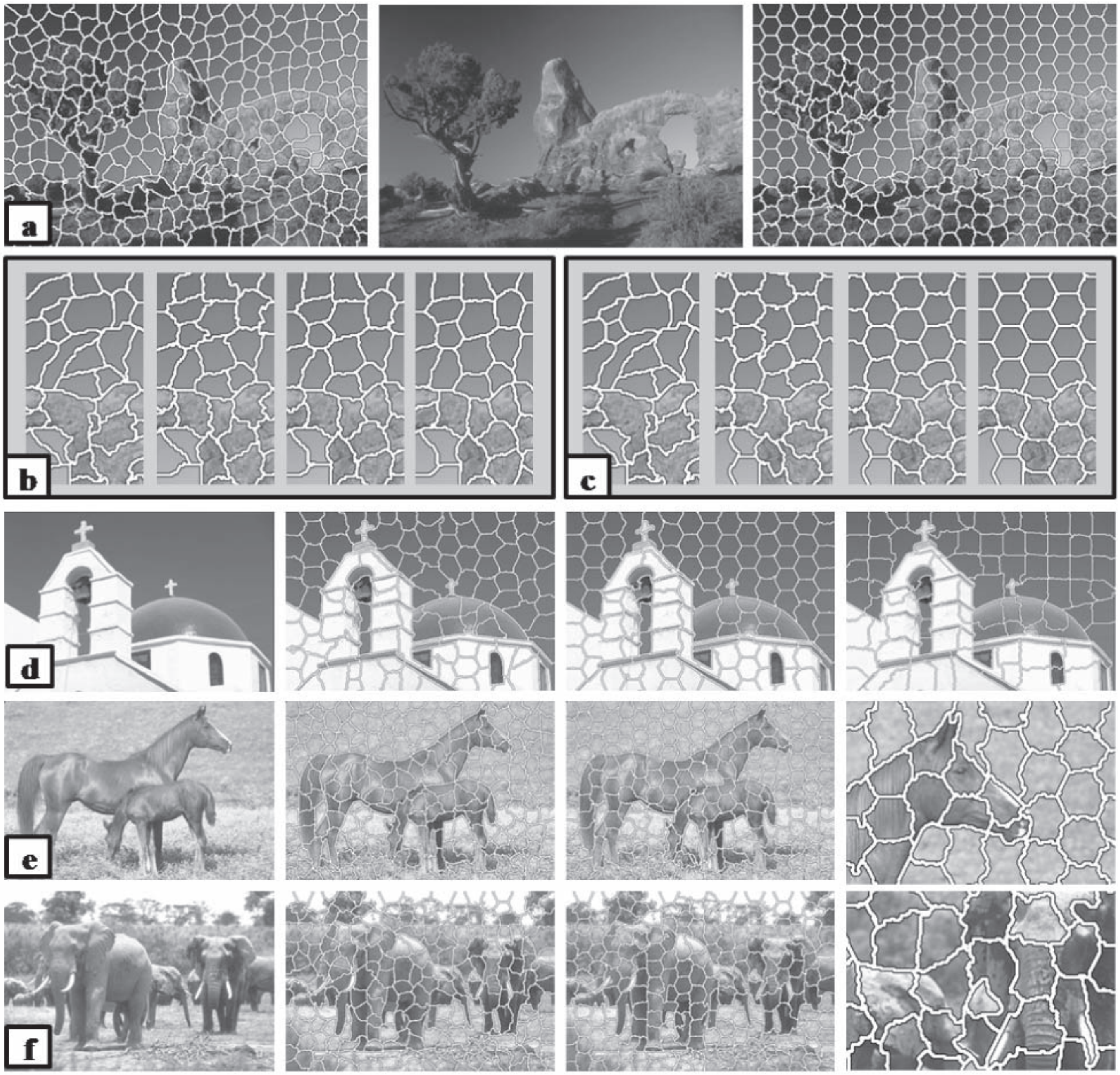


Fig. 3. Illustrations of waterpixels on the Berkeley segmentation database: All waterpixels images are computed with an hexagonal grid with step $\sigma = 30$ pixels and a regularization parameter $k = 8$, unless otherwise specified. (a): original image (middle) with corresponding *m-waterpixels* (left) and *c-waterpixels* (right). $\sigma = 25$ pixels, $k = 16$. (c): zooms of *m-waterpixels* (a) for $k = 0, 4, 8, 16$. (c): zooms of *c-waterpixels* (a) for $k = 0, 4, 8, 16$. (d): original image - *m-wat.* - *c-wat.* - *m-wat.* with square grid and $\sigma = 40$ pixels. (e): original image - *m-wat.* - *c-wat.* - zoom of *c-wat.*. (f): original image - *m-wat.* - *c-wat.* - zoom of *m-wat.* with $k = 16$.

image dependent template, this measure is more appropriate to evaluate regularity than compactness.

To quantify the adherence to object boundaries, a classical measure used in the literature is the boundary-recall (BR). Boundary-recall is defined as the percentage of ground-truth contour pixels GT which fall within strictly less than 3 pixels from superpixel boundaries C:

$$BR = \frac{|\{p \in GT, d(p, C) < 3\}|}{|GT|} \tag{6}$$

where d is the L_1 (or Manhattan) distance.

While precision cannot be directly used in the context of over-segmentations, boundary-recall has to be, in this particular case of superpixels, interpreted with caution. Indeed, as noted also by Kalinin and Sirota [24], very tortuous contours systematically lead to better performances: because of their higher number, SP contour pixels have a higher chance of matching a true contour, increasing artificially the boundary-recall. Hence, we propose to always consider the trade-off between boundary-recall and contour density to properly evaluate the adherence to object boundaries, penalizing at the same time the cost in pixels to describe SP contours.

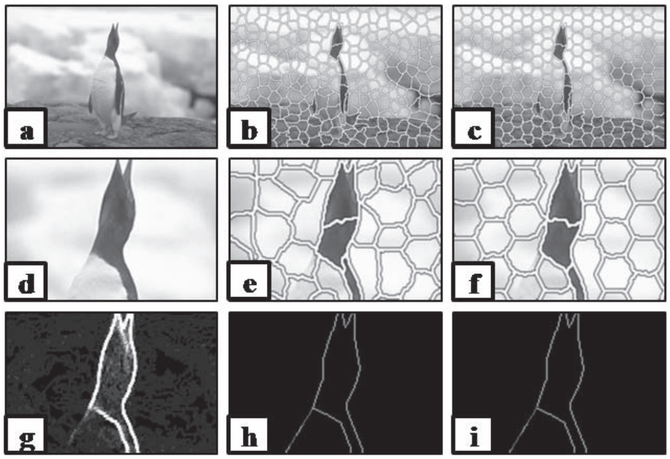


Fig. 4. Contours missed by waterpixels: (a): original image from the Berkeley segmentation database. (b): m-waterpixels with $step = 27$ and $k = 10$. (c): c-waterpixels with $step = 27$ and $k = 10$. (d), (e), (f): zoom of (a), (b), (c) respectively. (g): zoom of the non-regularized gradient image. (h) and (i): reached (green) and missed (red) contours, respectively by m-waterpixels and c-waterpixels.

D. Quantitative Analysis and Comparison With State-of-the-Art

In this paragraph, we will use m-waterpixels and denote them directly as "waterpixels" for the sake of simplicity.

During the design of the algorithm, we used intermediate results from the train and test subsets of the Berkeley database. Therefore, we report the results obtained for the validation subset ("val"), which contains 100 images. Results for boundary-recall, average mismatch factor and contour density are averaged for this subset and shown in Figure 5. Blue and red curves correspond to varying regularization parameters k and k' respectively for waterpixels and SLIC. The values for k and k' have been chosen such that they cover a reasonable portion of the regularization space between no regularization ($k = 0$) and a still acceptable level of regularization.

Figure 5(a) shows contour density against boundary-recall for waterpixels and SLIC. The ideal case being the lowest contour density for the highest boundary-recall, we can see that the trade-off between both properties improves for decreasing regularization, as expected. On the other hand, SLIC shows another behavior: the trade-off improves, then gets worse with regularization. At any rate, it is important to note that waterpixels achieves a better "best" trade-off than SLIC (see waterpixel $k = 0$ and SLIC $k' = 15$). Besides, this observation is valid for the whole family of waterpixel-methods as the zero-value regularization does not take into account d_Q. In order to do a fair comparison between waterpixels and SLIC over all criteria, we choose corresponding curves in the trade-off contour density/boundary-recall, *i.e.* waterpixels with $k = 8$ and SLIC with $k' = 15$, and compare this couple for the other criteria.

Figure 5(b) shows that, for a given number of superpixels, contour density of waterpixels is more stable and most of the time lower than SLIC when varying regularization. More particularly, contour density is lower for waterpixels ($k = 8$) than for SLIC ($k' = 15$). This means that for the same number of superpixels, waterpixels contours are shorter than SLIC contours, which is partly explained by less tortuous contours.

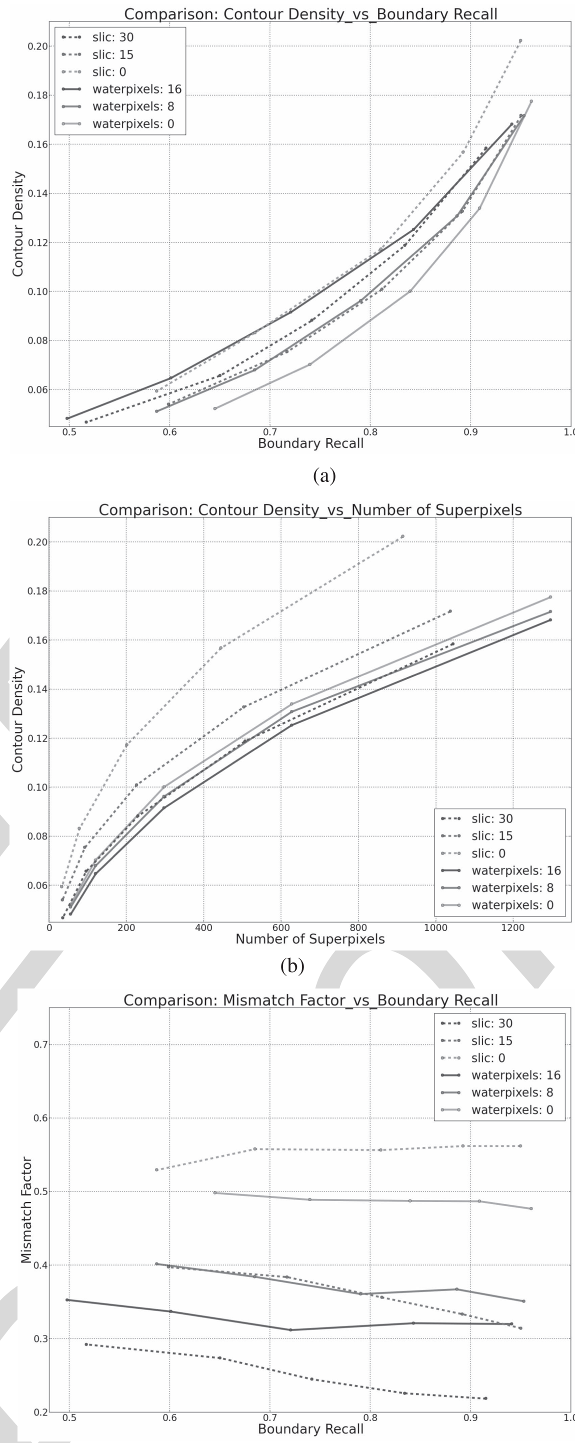


Fig. 5. Benchmark: performance comparison between waterpixels and SLIC. (a) Contour Density against Boundary-recall. (b) Contour Density against Number of Superpixels. (c) Mismatch factor against Boundary-recall.

Figure 5(c) shows average mismatch factor against boundary-recall for waterpixels and SLIC. We can see that the curves for waterpixels with $k = 8$ and SLIC with $k' = 15$ are here again close to each other.

(a) (b)

Fig. 6. Comparison between Waterpixels and SLIC superpixels for $\sigma = 25$ pixels on a zoom of an image from the Berkeley segmentation database. (a) SLIC $k' = 15$. (b) Waterpixels $k = 8$.

These properties are illustrated in Figure 6, where we can see examples of reached and missed contours by both methods, as well as their different behaviours in terms of regularity (shape, size, tortuosity).

E. Computation Time

Computing time was measured on a personal computer based on Intel(R) Core(TM) i7 central processing units (4 physical cores, 4 virtual ones), operating at 2.93GHz. Both methods have linear complexity with the number of pixels in the image. For an image of size 481×321, average computing time for SLIC was 149 ms, and 132 ms for waterpixels (82 ms without pre-filtering). A more detailed comparison of computation times is presented in Figure 7 (showing average and standard deviation for different numbers of superpixels). We can see that waterpixels are generally faster to compute than SLIC superpixels. Contrary to the latter's, their computation time decreases slightly with the number of superpixels. An analysis of computation times for the different steps of waterpixels reveals that this variability is only introduced by the grid computation and the minima selection procedure. Concerning grid computation time, it rises from 2 ms for small numbers of waterpixels to 27 ms for large numbers of waterpixels. This simply means that we still have to optimize this step. Concerning the computation time of the minima selection procedure, it decreases as waterpixels become larger because of pre-filtering step. Indeed, the size of this filtering is directly proportional to the cell size. As such, resulting images contain less minima, which simplifies the selection procedure. Besides, the variance observed when we change images is explained by the fact that the difficulty of minima evaluation/computation depends on the content of each image.

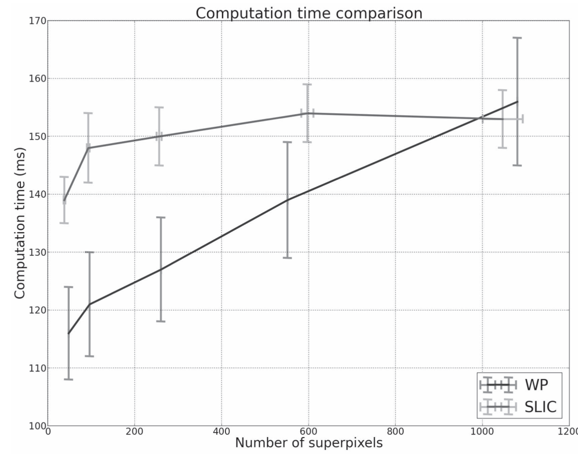


Fig. 7. Computation time comparison with images of the Berkeley database.

We are currently working on a new implementation of minima computation/evaluation which would be less dependent on the number of superpixels.

To conclude this section, waterpixels are generally faster to compute than SLIC superpixels, and they are at least as performant in the trade-off between adherence to object boundaries and regularity in shape and size, while using much less pixels to describe their contours.

V. Discussion and Perspectives

We have shown that waterpixels produce competitive results with respect to the state-of-the-art. These advantages are valuable in the classification/detection/segmentation pipeline, where superpixels play the part of primitives. Moreover, there is one major difference in the construction of the algorithm: the SLIC approach does not impose any connectivity constraint. The resulting superpixels are therefore not necessarily connected, which requires some *ad hoc* postprocessing step. In contrast, waterpixels are connected by definition, and the connectivity constraint is actually implemented in the distance used.

The proposed approach is gradient-based. Standard methods can be used to compute this gradient, or a specific gradient computation method can be designed for a given application. In any case, this offers flexibility to waterpixels. One limitation though is the quality of the signal in such a gradient image. As seen in 4, alteration by noise or insufficiently contrasted contours may lead to the prevalence of regularity over adherence to object boundaries. If filtering steps are usually enough to deal with noise and remove non pertinent small details, parameter values have to be optimized for each database. Future work will aim at overcoming this limitation by adding a learning step of optimal filtering values for specific databases.

The general design of waterpixels offers many prospects. Among them, one promising field of improvement resides in the placement of markers, as they constitute the main degree of freedom of the method. We are currently investigating the possibility to select the markers in an optimal manner, for example by formulating the marker placement as a p-dispersion problem (see [25]) in an augmented space.

The speed of waterpixels contributes to expanding their possible applications. For example, it could be interesting to compute different sets of waterpixels, by changing design options (different cells, gradients, grid steps, etc.), and then use ensemble clustering methods to obtain a final segmentation [26], [27].

Last buy not least, waterpixels lead to the efficient construction of hierarchical partitions based on superpixels. Indeed, the computation of the watershed can produce at the same time a segmentation and a hierarchy of partitions based on that segmentation, with only minor overhead computation times [28]–[30].

VI. Conclusion

This paper introduces waterpixels, a family of methods for computing regular superpixels based on the watershed transformation. Both adherence to object boundaries and regularity of resulting regions are encouraged thanks to the choice of the markers and the gradient to be flooded. Different design options, such as the distance function used to spatially regularized the gradient, lead to different trade-offs between both properties. The computational complexity of waterpixels is linear. Our current implementation makes it one of the fastest superpixel methods. Experimental results show that waterpixels are competitive with respect to the state-of-the art. They outperform SLIC superpixels, both in terms of quality and speed. The trade-off between speed and segmentation quality achieved by waterpixels, as well as their ability to generate hierarchical segmentations at negligible extra cost, offer interesting perspectives for this superpixels generation method.

An implementation of waterpixels is available from http://cmm.ensmp.fr/~machairas/waterpixels.

Appendix
Mean Mismatch Factor Definition

Let $\{s_i\}_{1\leq i\leq N}$ be a set of superpixels. The centered version s_i^* of s_i is obtained by translating s_i so that its barycenter is the origin of the coordinates system.

The average shape $\widehat{s}^*$ of the $\{s_i\}$ is computed as follows. Let first define function S:

$$S:\left|\begin{array}{l} D \longrightarrow \mathbb{N} \\ x_p \longmapsto \sum_{i=1}^{N} 1_i(x_p) \end{array}\right. \tag{7}$$

where 1_i is the indicator function of s_i^*. Thus, image S corresponds to the summation image of all centered superpixels. Let furthermore $\mu_A = 1/n\sum_{i=1}^{N}|s_i|$ be the average area of the considered superpixels, and let S_t be the threshold of S at level t: $S_t(x) = \{x_p \in D \,\big|\, |S(x_p)| \geq t\}$.

The average centered shape $\widehat{s}^*$ is then the set S_{t_0}, where t_0 is the maximal threshold value which enables $\widehat{s}^*$ to have an area greater than or equal to μ_A:

$$t_0 = \max\{t \,\big|\, |S_t| \geq \mu_A\} \tag{8}$$

$$\widehat{s}^* = S_{t_0} \tag{9}$$

Finally, the mean mismatch factor of superpixels $\{s_i\}_{1\leq i\leq N}$ is:

$$MF = \frac{1}{N}\sum_{i=1}^{N} mf(s_i^*, \widehat{s}^*). \tag{10}$$

References

[1] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection," in *Proc. Int. Workshop Image Process., Real-Time Edge Motion Detection/Estimation*, 1979. AQ:5

[2] S. Beucher and F. Meyer, "The morphological approach to segmentation: The watershed transformation," in *Mathematical Morphology in Image Processing*, E. Dougherty, Ed. 1993, pp. 433–481. AQ:6

[3] F. Meyer, "Un algorithme optimal pour la ligne de partage des eaux," *Dans 8^e Congrés Reconnaissance Formes Intell. Artif.*, vol. 2, pp. 847–857, Nov. 1991.

[4] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[5] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[6] V. Machairas, E. Decencière, and T. Walter, "Waterpixels: Superpixels based on the watershed transformation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 4343–4347.

[7] O. Monga, "An optimal region growing algorithm for image segmentation," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 1, nos. 3–4, pp. 351–375, 1987. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218001487000242

[8] B. Marcotegui and F. Meyer, "Bottom-up segmentation of image sequences for coding," *Ann. Télécommun.*, vol. 52, nos. 7–8, pp. 397–407, 1997. [Online]. Available: http://link.springer.com/article/10.1007/BF02998459

[9] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1. Oct. 2003, pp. 10–17.

[10] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[11] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[12] J. Wang and X. Wang, "VCells: Simple and efficient superpixels using edge-weighted centroidal Voronoi tessellations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1241–1247, Jun. 2012.

[13] G. Zeng, P. Wang, J. Wang, R. Gan, and H. Zha, "Structure-sensitive superpixels via geodesic distance," *Int. Conf. Comput. Vis.*, vol. 1, no. 1, pp. 1–27, 2011.

[14] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 211–224.

[15] P. Soille, *Morphological Image Analysis: Principles and Applications*. New York, NY, USA: Springer-Verlag, 2003.

[16] B. Andres, U. Köthe, M. Helmstaedter, W. Denk, and F. A. Hamprecht, "Segmentation of SBFSEM volume data of neural tissue by hierarchical classification," in *Pattern Recognition*, Berlin, Germany: Springer-Verlag, 2008, pp. 142–152.

[17] J. Stawiaski, E. Decencière, and F. Bidault, "Interactive liver tumor segmentation using graph cuts and watershed," in *Proc. MICCAI*, New York, NY, USA, 2008. AQ:7

[18] P. Neubert and P. Protzel, "Compact watershed and preemptive SLIC: On improving trade-offs of superpixel segmentation algorithms," in *Proc. IEEE 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 996–1001.

[19] C. Vachier and F. Meyer, "Extinction values: A new measurement of persistence," in *Proc. IEEE Workshop Non Linear Signal/Image Process.*, 1995, pp. 254–257.

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, vol. 2. Jul. 2001, pp. 416–423.

[21] M. Faessel and M. Bilodeau, "SMIL: Simple morphological image library," LRDE, Tech. Rep., 2013. AQ:8

[22] N. J. C. Strachan, P. Nesvadba, and A. R. Allen, "Fish species recognition by shape analysis of images," *Pattern Recognit.*, vol. 23, no. 5, pp. 539–544, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/003132039090074U

[23] A. Schick, M. Fischer, and R. Stiefelhagen, "An evaluation of the compactness of superpixels," *Pattern Recognit. Lett.*, vol. 43, pp. 71–80, Jul. 2014.

[24] P. Kalinin and A. Sirota, "A graph based approach to hierarchical image over-segmentation," *Comput. Vis. Image Understand.*, vol. 130, pp. 80–86, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314214001891

[25] E. Erkut, "The discrete p-dispersion problem," *Eur. J. Oper. Res.*, vol. 46, no. 1, pp. 48–60, May 1990.

[26] K. Cho and P. Meer, "Image segmentation from consensus information," *Comput. Vis. Image Understand.*, vol. 68, no. 1, pp. 72–89, Oct. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314297905464

[27] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003. [Online]. Available: http://dx.doi.org/10.1162/153244303321897735

[28] F. Meyer, "Minimum spanning forests for morphological segmentation," in *Mathematical Morphology and Its Applications to Image Processing*. Boston, MA, USA: Kluwer, Sep. 1994, pp. 77–84.

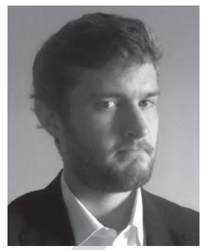
[29] S. Beucher, "Watershed, hierarchical segmentation and waterfall algorithm," in *Mathematical Morphology and Its Applications to Image Processing*, J. Serra and P. Soille, Eds. Fontainebleau, France: Kluwer, Sep. 1994, pp. 69–76.

[30] F. Meyer, "An overview of morphological segmentation," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 7, pp. 1089–1118, 2001.

Vaïa Machairas received the Engineering degree in optics from the Institut d'Optique Graduate School (Supoptique), Palaiseau, France, and the master's degree in optics, image, vision from Jean Monnet University, Saint Etienne, France, both in 2013. She is currently pursuing the Ph.D. degree with the Centre for Mathematical Morphology, MINES ParisTech. Her research interests include mathematical morphology, image segmentation, machine learning, and colorimetry.

Matthieu Faessel received the Ph.D. degree in engineer sciences from the University of Bordeaux, France, in 2003. He is currently a Research Engineer with the Centre of Mathematical Morphology, School of Mines, Paris, France. His research interests include image segmentation, computer vision, and materials.

David Cárdenas-Peña received the bachelor's degree in electronic engineering and the M.Eng. degree in industrial automation from the Universidad Nacional de Colombia, Manizales-Colombia, in 2008 and 2011, respectively. He is currently pursuing the Ph.D. degree in automatics with the Universidad Nacional de Colombia. He has been a Research Assistant with the Signal Processing and Recongnition Group since 2008. His current research interests include machine learning and signal and image processing.



Théodore Chabardes received the degree from the Engineering School, ESIEE Paris, France, in 2014, as an Engineer specialized in computer science. He is currently pursuing the Ph.D. degree with the Centre of Mathematical Morphology, School of Mines, Paris, France. His research interests include mathematical morphology, image segmentation, and software optimization.

Thomas Walter received the Diploma degree in electrical engineering from Saarland University, Germany, and the Ph.D. degree in mathematical morphology from Mines ParisTech, France. He held a post-doctoral position with the European Molecular Biology Laboratory, Heidelberg, Germany. He is currently a Team Leader in bioimage informatics with the Centre for Computational Biology, Mines ParisTech, and a member of the Bioinformatics Unit with the Curie Institute, Paris. His most visible scientific contributions have been in the field of bioimage informatics, and in particular, in high content screening. He has pioneered methods in the field of cellular phenotyping and phenotypic clustering from live cell-imaging data. He was involved in the first genome-wide screen by live cell imaging in a human cell line, and co-develops the open-source software cellcognition.

Etienne Decencière received the Engineering degree and the Ph.D. degree in mathematical morphology from MINES ParisTech, France, in 1994 and 1997, respectively, and the Habilitation à Diriger des Recherches from Jean Monnet University, in 2008. He holds a research fellow position with the Centre for Mathematical Morphology, MINES ParisTech, where he leads several academic and industrial research projects. His main research interests are in mathematical morphology, image segmentation, and biomedical applications.

AUTHOR QUERIES

AQ:1 = Please check whether the edits made in the financial section are OK.
AQ:2 = Please confirm the current affiliation of all authors.
AQ:3 = Please confirm the postal code for "MINES ParisTech, PSL Research University, Center for Mathematical Morphology, Universidad Nacional de Colombia, Centre for Computational Biology, Institut Curie, and Inserm."
AQ:4 = Table I is not cited in body text. Please indicate where it should be cited.
AQ:5 = Please provide the page range for ref. [1].
AQ:6 = Please provide the publisher name and location for ref. [2].
AQ:7 = Please provide the page range and also confirm the conference title for ref. [17].
AQ:8 = Please provide the organization, location, and report no. for ref. [21].